Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Constructive learning neural network applied to identification and control of a fuel-ethanol fermentation process

Luiz Augusto da Cruz Meleiro [a,*], Fernando José Von Zuben [b], Rubens Maciel Filho [c]

[a] Department of Food Engineering, Federal Rural University of Rio de Janeiro, Caixa Postal 74573, CEP 23890-971 Seropedica, RJ, Brazil
[b] School of Electrical and Computer Engineering, State University of Campinas, FEEC-UNICAMP, Caixa Postal 6101, CEP 13083-970 Campinas, SP, Brazil
[c] School of Chemical Engineering, State University of Campinas, FEQ-UNICAMP, Caixa Postal 6066, CEP 13081-970 Campinas, SP, Brazil

ARTICLE INFO

Article history:
Received 22 June 2007
Received in revised form
30 April 2008
Accepted 22 June 2008
Available online 30 August 2008

Keywords:
Model predictive control
Constructive neural networks
Fermentation process
Bioprocess identification
Dynamic simulation

ABSTRACT

In the present work, a constructive learning algorithm was employed to design a near-optimal one-hidden layer neural network structure that best approximates the dynamic behavior of a bioprocess. The method determines not only a proper number of hidden neurons but also the particular shape of the activation function for each node. Here, the projection pursuit technique was applied in association with the optimization of the solvability condition, giving rise to a more efficient and accurate computational learning algorithm. As each activation function of a hidden neuron is defined according to the peculiarities of each approximation problem, better rates of convergence are achieved, guiding to parsimonious neural network architectures. The proposed constructive learning algorithm was successfully applied to identify a MIMO bioprocess, providing a multivariable model that was able to describe the complex process dynamics, even in long-range horizon predictions. The resulting identification model was considered as part of a model-based predictive control strategy, producing high-quality performance in closed-loop experiments.

1. Introduction

Artificial neural networks (ANN) have been widely applied to the identification and control of nonlinear dynamic systems (Ng, 1997). One of the main reasons for this success is the universal approximation capability of ANN, i.e., such models are able to approximate to arbitrary accuracy any continuous mapping defined on a compact (closed and bounded) domain (Jones, 1987; Cybenko, 1989; Hornik, 1989). However, due to their generic structure, neural models usually require the estimation of a large number of parameters. Problems related to computational procedures necessary to achieve good results, including definition of the neural network dimension, choice of nonlinear activation functions, and the search for the optimum weight set, are still a drawback to a wider use of ANN (Kosko, 1997; Haykin, 1999; Dote and Ovaska, 2001). Another aspect to be considered is the generalization capability associated with supervised learning techniques when applied to universal approximators (Geman et al., 1992).

To get around these problems, Von Zuben and Netto (1995) presented a unit-growing learning (UGL) approach that corresponds to the projection pursuit learning (PPL) in association with the optimization of the solvability condition. The improved PPL is a constructive learning algorithm (Kwok and Yeung, 1997) characterized by a more efficient and accurate computational procedure for non-parametric regression (Von Zuben and Netto, 1997). The solvability condition states that a neural network with one nonlinear hidden layer and one linear output layer is theoretically able to learn any input–output continuous mapping, given that the number of nodes of the hidden layer is capable of reproducing the dimensionality of the input space.

Model predictive control (MPC) algorithms have been widely used in industrial processes in recent years (Henson, 1998). These algorithms are well suited for high performance control of constrained multivariable processes, mainly because explicit pairing of input and output variables is not required, and constraints can be directly incorporated into the controller design (Henson, 1998). On the other hand, most of the industrial applications use linear dynamic models (Qin and Badgwell, 1997; Piché et al., 2000). Although the use of nonlinear models may improve the control algorithm performance in a rate not achievable by linear approaches, the development of such models is usually associated with challenging design tasks (Piché et al., 2000).

In recent years, there has been a strong interest in the use of neural networks to describe chemical processes, due to their ability to approximate highly nonlinear systems. Different

* Corresponding author. Tel.: +55 21 3787 3750; fax: +55 21 3787 3742.
E-mail addresses: meleiro@ufrrj.br (L.A.C. Meleiro), vonzuben@dca.fee.unicamp.br (F.J. Von Zuben), maciel@feq.unicamp.br (R.M. Filho).

0952-1976/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2008.06.001

architectures of neural networks have been used as nonlinear models to advanced control algorithms (Zhan and Ishida, 1997; Qin and Badgwell, 1998; Rohani et al., 1999; Kambhampati et al., 2000). Hussain (1999) presented a review of neural network applications in process control, pointing out that these nonlinear input–output models are capable of identifying the system in a great number of cases, and can be incorporated into various well-known nonlinear control methods.

In spite of the existence of several proposals in the literature concerning neural network architectures as identification models for predictive control, optimization procedures for the automatic definition of the neural network dimension, and for the choice of the best set of activation functions for the hidden neurons are still deserving a deeper attention, mainly because a successful solution to both optimization problems will certainly guide to a parsimonious identification model.

The case study considered here is an extractive alcoholic fermentation process (Silva et al., 1999). One of the options to improve the productivity of this process is the continuous extraction of ethanol. Several schemes combining fermentation with a separation process have been proposed (Costa et al., 2001). For instance, Silva et al. (1999) have shown that a scheme combining a fermenter with a vacuum flash vessel presents several positive features and better performance than an industrial conventional process. Another important aspect to be considered in the optimization of the alcoholic fermentation process is the development of an efficient control strategy, since it can minimize costs and environmental impact by maintaining the process under optimum conditions. However, biotechnological processes are characterized by their complex dynamics, such as inverse response, dead time and strong nonlinearities. For these reasons, modeling and control of those systems are still important problems that have to be solved.

In the present work, the flexibility of the UGL approach (Von Zuben and Netto, 1995, 1997)—that simultaneously addresses the proper definition of the neural network dimension, the choice of nonlinear activation functions, and the search for the optimum weights set—is properly explored and applied as an effective tool to model and control the extractive alcoholic fermentation process developed by Silva et al. (1999).

The main objective of this work was to develop a MIMO UGL neural model-based predictive control algorithm for the aforementioned extractive alcoholic fermentation process (Silva et al., 1999). In previous works, Meleiro et al. (2002, 2003, 2005, 2007) have already investigated the capability of advanced nonlinear approaches for process identification and predictive control, though restricted to a simpler control configuration for this fermentation process (Costa et al., 2002; Meleiro et al., 2005), or considering a different fermentation process. The control methodology proposed here considers constraints on manipulated and controlled variables. Moreover, it has an identification model providing high-quality long-range horizon predictions, and it is capable of dealing with unmodeled load disturbances. The associated predictive control algorithm uses the successive quadratic programming (SQP) method to solve the optimization problem at each sample interval.

The paper is organized as follows: Section 2 presents the constructive learning algorithm (UGL) employed to design a proper one-hidden layer neural network structure that best approximates a given mapping. The case study, an extractive alcoholic fermentation process, is detailed in Section 3. The results of the proposed modeling and control schemes applied to this process are discussed in Section 4. The results of the identification task provided by the UGL neural models for this process are shown in Section 4.1, and the proposed predictive control scheme based on this neural model, as well as its closed-loop performance, are presented and discussed in Section 4.2. Finally, concluding remarks are addressed in Section 5.

2. Unit-growing learning

Although the problems related to a proper estimation of the set of connection weights in conventional neural networks have been solved with relative success by using advanced first- and second-order nonlinear optimization algorithms, the remaining two obstacles, i.e., definition of the neural network dimension and choice of the activation functions, are still solved heuristically in most applications by exhaustive trial and error procedures (Battiti, 1992). Bärmann and Biegler-König (1992) presented a kind of unit-growing method that, along the learning process, enables additional neurons to be incorporated into the hidden layer by means of an iterative procedure, until a proper network dimension (number of nodes) is achieved. However, this method uses monotonic functions determined arbitrarily, which does not guarantee the optimality of the set of activation functions. Hwang et al. (1994) successfully applied parametric models using an orthonormal set of basis functions to solve the problem of searching for optimal activation functions. Von Zuben and Netto (1995) discussed the solvability condition applied to constructive learning in neural networks, and developed an iterative procedure that conciliates nonlinear optimization, UGL and parametric activation function modeling, aiming at generating optimal neural network configurations for MIMO mappings.

2.1. The UGL method

Consider the regression problem whose objective is to generate the best approximation of an unknown multidimensional model-free continuous function $G(\cdot)$ defined as

$$G(\cdot) : \Re^{1\times m} \to \Re^{1\times r} \tag{1}$$

Starting from N pairs of input–output vectors extracted from the unknown multidimensional mapping of Eq. (1), and considering additive error such that

$$\boldsymbol{s}_l = G(\boldsymbol{x}_l) + \boldsymbol{\varepsilon}_l \tag{2}$$

we obtain, for $l = 1, \ldots, N$:

$$(\boldsymbol{x}_l, \boldsymbol{s}_l) = ([x_{l1} \;\; x_{l2} \;\; \cdots \;\; x_{lm}], [s_{l1} \;\; s_{l2} \;\; \cdots \;\; s_{lr}]) \tag{3}$$

The goal of the regression task is to construct an estimator, $\hat{G}(\cdot)$, that provides the best approximation of $G(\cdot)$ in such a way that it is able to predict $\boldsymbol{s}_t$, given $\boldsymbol{x}_t$ $(t \neq l)$, as follows:

$$\hat{\boldsymbol{s}}_t = \bar{\boldsymbol{s}} + \hat{G}(\boldsymbol{x}_t) \tag{4}$$

where $\bar{\boldsymbol{s}} \in \Re^{1\times r}$ is the sample average over all the desired output training data, given by

$$\bar{\boldsymbol{s}} = [\bar{s}_1 \;\; \bar{s}_2 \;\; \cdots \;\; \bar{s}_r] = \frac{1}{N}\sum_{l=1}^{N} \boldsymbol{s}_l \tag{5}$$

One-hidden layer neural networks can be used to estimate the matrix of output response, $\boldsymbol{S} \in \Re^{N\times r}$, given the matrix of independent variables, $\boldsymbol{X} \in \Re^{N\times m}$, as follows:

$$\hat{s}_{lk} = \bar{s}_k + \sum_{j=1}^{n}\left[w_{jk} f_j\left(\sum_{i=1}^{m} v_{ij} x_{li}\right)\right] \quad \text{with} \quad \begin{array}{l} k = 1, \ldots, r \\ l = 1, \ldots, N \end{array} \tag{6}$$

where v_{ij} $(\boldsymbol{V} \in \Re^{m\times n})$ denotes the hidden layer weights connecting the i-th element of the input to the j-th hidden node, w_{jk} $(\boldsymbol{W} \in \Re^{n\times r})$ denotes the output layer weights connecting the j-th hidden node to the k-th output node, and $f_j : \Re \to \Re$ is the trainable activation function of the j-th hidden node.